

Received January 8, 2022, accepted January 25, 2022, date of publication February 4, 2022, date of current version February 14, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3149050

Design and Implementation of Traffic Generation Model and Spectrum Requirement Calculator for Private 5G Network

DAEGYEOM KIM¹, MYEONGJIN KO¹, SUNGHYUN KIM¹, SUNGWOO MOON¹,
KYUNG-YUL CHEON², SEUNGKEUN PARK², (Member, IEEE), YUNBAE KIM²,
HYUNGOO YOON³, AND YONG-HOON CHOI¹, (Member, IEEE)

¹School of Robotics, Kwangwoon University, Nowon-gu, Seoul 01897, South Korea

²Electronics and Telecommunications Research Institute (ETRI), Yuseong-gu, Daejeon 34129, South Korea

³Department of Electronic Engineering, Myongji College, Seodaemun-gu, Seoul 03656, South Korea

Corresponding authors: Hyungoo Yoon (hgyoon@mjc.ac.kr) and Yong-Hoon Choi (yhchoi@kw.ac.kr)

This work was supported in part by the Institute for Information & Communications Technology Promotion (IITP) Grant funded by the Korea Government, Ministry of Science and ICT (MSIT) (Development of Technology for Securing and Supplying Radio Resources), under Grant 2021-0-00092; in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government MSIT under Grant 2021R1F1A1064080; and in part by the Research Grant of Kwangwoon University, in 2021.

ABSTRACT This paper proposes a neural 5G traffic generation model and a methodology for calculating the spectrum requirements of private 5G networks to provide various industrial communication services. To accurately calculate the spectral requirements, it is necessary to analyze the actual data volume and traffic type of industrial cases. However, because there is currently no suitable traffic model to test loads in private 5G networks, we have developed a generative adversarial network (GAN)-based traffic generator that can generate realistic traffic by learning actual traffic traces collected by mobile network operators. In addition, in the case of industrial applications, probability-based traffic models were used in parallel as there were not enough real data to be learned. The proposed 5G traffic generation model is combined with the proposed 5G spectrum calculation methodology, enabling more accurate spectrum requirements calculation through traffic simulation similar to a real-life environment. In this paper, the spectrum requirements are calculated differently according to two types of duplexing, namely frequency division duplexing (FDD) and time division duplexing (TDD). As a guide for companies aiming to provide advanced wireless connectivity for a wide variety of vertical industries using 5G networks, eight use cases defined in the 5G Alliance for Connected Industries and Automation (ACIA) white paper were simulated. The spectrum requirements were calculated under various simulation conditions considering varying traffic loads, deployment scenarios, and duplexing types. Various simulation results confirmed that a bandwidth of at least 22.0 MHz to a maximum of 397.8 MHz is required depending on the deployment scenario.

INDEX TERMS Generative adversarial network (GAN), deep learning, 5G traffic, real traffic trace, spectrum requirement calculation, private 5G network, industrial use cases.

I. INTRODUCTION

According to Allied Business Intelligence (ABI) Research's announcement in July 2020, the scale of investment in 5G private networks is expected to grow rapidly, reaching 24 billion dollars in 2035 beyond 5G public networks. In the beginning,

The associate editor coordinating the review of this manuscript and approving it for publication was Ting Wang¹.

private 5G network services using the carrier's frequency will lead to growth, but it is expected that it will gradually shift towards building private 5G networks with local 5G frequencies. When constructing a private network, it is important to accurately predict the amount of traffic demand. Since the frequency requirement or the capacity of the network device is determined based on the traffic load, traffic source models or actual traffic measurement data are required to predict the

load on the network. However, there is currently no traffic generation model suitable for traffic on private 5G networks, and there are not enough real traffic trace data. Datasets measured in real networks are publicly available [1], [2], but as these were measured under specific circumstances, it is difficult to use them to determine the spectrum requirements or equipment capacity of a private 5G network applicable to various use cases. To estimate the spectral requirements of a private 5G network, we first determined a deployment scenario based on actual industrial use cases, and then determined the traffic models and related parameters to apply to each use case to generate traffic; finally, the frequency bandwidth required to accommodate the generated traffic was calculated.

Existing stochastic traffic source models such as the interrupted Bernoulli process (IBP) and Markov modulated Poisson process (MMPP) are not suitable for modeling recent web-based video traffic such as Netflix streaming service or Zoom video conferencing. In addition, the traffic model that simulates the packet generation pattern of a codec (e.g., H.263 and H.264) is suitable as a source model for a video server but has limitations in simulating the download traffic pattern of a subscriber network. As the appearance of the download traffic of the subscriber network varies greatly according to the configuration of the application server of the subscriber network, load balancer, and firewall, it is more ideal to generate traffic based on measured data rather than a mathematical traffic source model. The proposed traffic model is a generative neural network model capable of generating or synthesizing traffic similar to real data by learning a real 5G dataset.

Typical spectrum requirements calculation methodologies predict traffic demand based on market research or traffic models and convert them into spectrum requirements. Standard recommendations for calculating spectrum requirements include ITU-R M.1390 [3] applied to circuit-switched networks, and ITU-R M.1768-1 [4] applied to a packet-switched network. The former computes the traffic demand using a simple Erlang-B formula, whereas the latter computes the traffic demand based on the M/G/1 queuing model, which can reflect the statistical characteristics of the packet and priorities of service classes. The authors of [5] calculated the aeronautical mobile airport communication system (AeroMACS) spectrum requirements using the methodology presented in [4]. Kim and Park [6] presented a mathematical approach for calculating the spectrum requirements of 5G enhanced mobile broadband (eMBB) and ultra-reliable and low-latency communication (URLLC) based on the M/G/1 queuing model with the same input parameters as [4]. However, both methodologies typically require statistical characteristics of offered traffic load from market research or a stochastic model for the packet arrival process. For cellular networks operated by mobile network operators (MNOs) with a large number of users worldwide, traffic data can be obtained through market research, but for some applications such as smart factories and smart farms, which have relatively fewer users and limited usage, it is difficult to obtain traffic

data through the same channels. Therefore, in this paper, we developed a 5G traffic simulator including the existing probability models and the proposed neural network-based traffic generation model. In addition, this paper proposes a new 5G spectrum requirement calculation method that does not use either the existing standard methodology [3], [4], nor queuing theory.

The remainder of this paper proceeds as follows. Section II describes related works. Section III describes the proposed generative neural network model, the dataset used for training, and how to train it to generate 5G traffic. Section IV describes the private 5G spectrum calculation methodology based on generated traffic. Section V describes use cases and scenarios for calculating spectrum requirements. Section VI describes the performance evaluation of the proposed technique in terms of 5G traffic generation and spectrum estimation. Finally, Section VII concludes this paper.

II. RELATED WORKS

Research on packet traffic source models for audio and video has been in progress for a long time. After the two-state model with silent and talk spurt state was proposed by Brady [7], research on packet voice modeling was actively conducted until the early 2000s. Three major packet traffic source models (i.e., the MMPP model, semi-Markov process (SMP) model, and fluid flow model) were analyzed and compared by Daigle and Langford [8]. In the late 1980s, with the emergence of moving picture experts group (MPEG) standards and asynchronous transfer mode (ATM) networks capable of carrying video traffic, several traffic models incorporating codec characteristics of variable bit rate (VBR) compressed video were proposed [9]–[11]. A video traffic model is a stochastic model that generates the size of each successive encoded video frame. In general, the parameters of the model are statistically obtained by analyzing a given frame trace. Self-similar properties that appear in local area network (LAN) data traffic and transmission control protocol (TCP) traffic also appear in video traffic [12]. Recently, multiview coding (MVC) video and three-dimensional (3D) video traffic models have also been conducted Tanwir *et al.* [13] proposed a 3D video traffic model based on a Markov modulated gamma process (3D-MMG) and compared the model with the hidden Markov model (HMM).

The overall discussion of the recent stochastic traffic source model is summarized in [14]–[16]. Most of the packet traffic simulators [17], [18] currently used in academia and industry are based on stochastic traffic source models. These stochastic models simulate the packet generation pattern of a codec and are, therefore, suitable as a source model for a video server but have limitations in simulating the download traffic pattern of a subscriber network. For example, the recent web-based video streaming traffic is greatly affected by the request cycle of the video client, regardless of the codec's traffic generation pattern. Fig. 1(a) and 1(b) show the downlink traffic patterns of Amazon Prime Video and Netflix, respectively. As shown in Fig. 1, when users watch

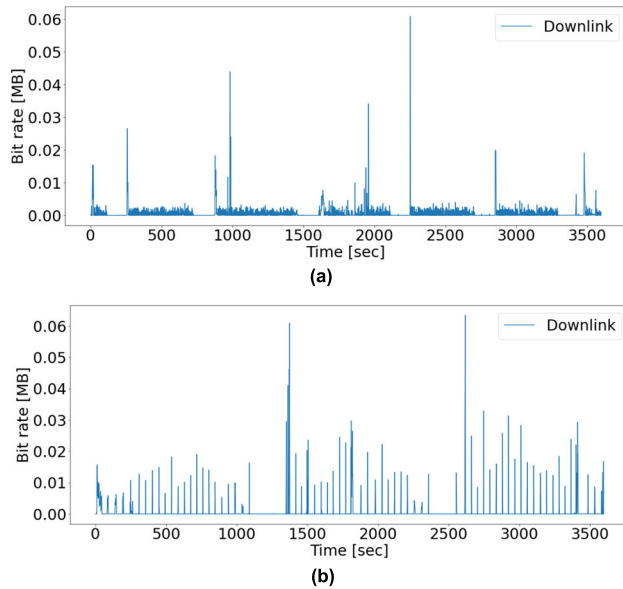


FIGURE 1. Measured downlink video streaming traffic patterns. (a) Amazon Prime Video. (b) Netflix

the same video content using different over-the-top (OTT) services, it is evident that the download traffic patterns are different. This is because clients have different request cycles. Therefore, it is difficult to simulate downlink traffic similarly to the actual situation using existing video source models.

Over the last several years, a variety of studies on short-term prediction models based on measured data have been conducted. In particular, time series analysis has been widely used for traffic load prediction. Time series analysis observes the measured dataset and extracts statistical characteristics to derive a statistical model suitable for these characteristics. The most widely used time series traffic prediction model is the autoregressive integrated moving average (ARIMA) model. Kim *et al.* [19] performed packet traffic forecasting using the ARIMA model with various datasets downloaded from the National Laboratory for Applied Network Research (NLANR) [20] with various traffic sources and sampling intervals. Otoshi *et al.* [21] applied seasonal ARIMA models to predict short- and long-term future traffic volumes. Guo *et al.* [22] applied a seasonal multiplication ARIMA model to predict mobile communication traffic, and Chen *et al.* [23] applied a seasonal ARIMA model to predict IEEE 802.11 traffic. In addition to ARIMA, various time series analysis models have been applied to network traffic analysis, and it is possible to express not only one VBR video source model but also a model in which several sources are combined. However, the time series analysis method is limited in that it requires prior analysis of the autocorrelation function (ACF) and partial autocorrelation function (PACF) to determine appropriate model parameters. The statistical model has the advantage of lower computational complexity compared to machine learning (ML) techniques and is thus, suitable for the short-term prediction of gently changing

traffic based on already analyzed datasets. However, because 5G traffic loads measured by base stations or terminals often have burstiness and long-term correlation characteristics, it is difficult to accurately predict the traffic load using statistical models.

Recently, as ML techniques based on artificial neural networks are widely used, studies that predict traffic volume by learning traffic traces are emerging. The most widely used ML models for time series forecasting are recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU). Fan *et al.* [24] proposed a neural network-based network traffic prediction model that combines a deep RNN and a GRU applied to predict network traffic. The results were verified by numerical calculation and simulation, and the network traffic prediction results of the model were found to be close to the real values. Xu *et al.* [25] proposed a deep neural network for traffic prediction of a cellular network. They use LSTM with an attention mechanism to encode the long-term dependencies of the traffic, and then decode the obtained state by the time convolutional network (TCN). They compared the proposed model with several ML models, including the seasonal ARIMA model. To predict 5G mobile traffic, the authors of [26] presented a state-of-the-art ML framework based on graph convolutional networks (GCNs). They considered practical constraints in prediction mechanisms such as limited data availability and lack of recent measurements. Recurrent neural network families, such as GRU and LSTM, show excellent predictive performance when trained properly while avoiding gradient vanishing problems and overfitting, but are not suitable for use as traffic generators because it is difficult to continuously generate traffic.

Cheng [27] proposed and prototyped a generative adversarial network (GAN) model for generating realistic network traffic at the Internet protocol (IP) packet level. The use of GANs for generating network packets is novel compared to existing flow-based GAN models [28], [29]. A new technique for encoding network data specifically for use in a convolutional neural network (CNN)-based generator was introduced. Shahid *et al.* [30] proposed combining an autoencoder with a GAN to generate sequences of packet sizes that correspond to bidirectional flows. The autoencoder was trained to learn a latent representation of the real sequences of packet sizes and a GAN was then trained on the latent space, to learn to generate latent vectors that can be decoded into realistic sequences. This approach generates sequences of packet sizes that behave closely to real Internet of things (IoT) bidirectional flows.

Currently, two 5G datasets are publicly available [1], [2]. The first dataset [1] is 65 KB in size and was collected for 5G network slicing research [31], [32]. The second dataset [2] is a 5G trace dataset collected from a major Irish mobile network operator. The dataset [2] has two mobility patterns (i.e., static and car) and two application patterns (i.e., video streaming and file download). The dataset is composed of client-side cellular key performance indicators (KPIs) comprising

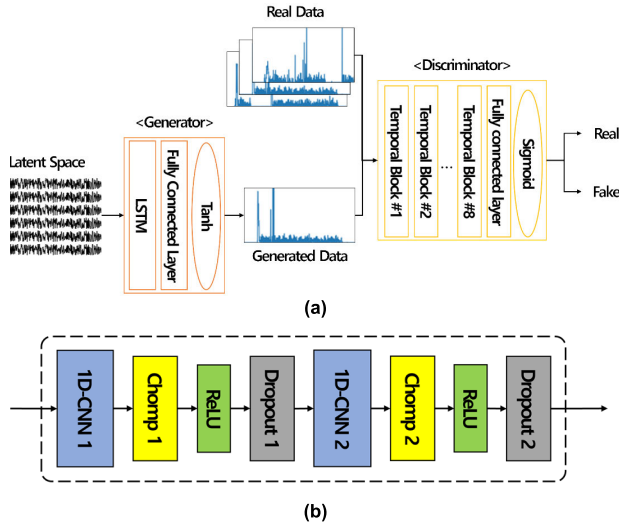


FIGURE 2. (a) The proposed 5G Traffic-GAN model architecture. (b) Temporal block structure of discriminator.

channel-related metrics, context-related metrics, cell-related metrics, and throughput information. These metrics were created using G-NetTrack Pro, a well-known and unrooted Android network monitoring application.

III. TRAFFIC GENERATION SCHEME

A. RECENT ADVANCES IN GAN

Originally proposed by Goodfellow *et al.*, the GAN [33] is an unsupervised deep learning machine, where the generator learns how to mimic the target data distribution, while the discriminator tries to differentiate between the real data and the samples coming from the generator. GAN models are developing rapidly and have achieved remarkable success in image and video synthesis. Recent image generation models have greatly improved the visual fidelity and resolution of the generated images [34]–[36]. Conditional GAN [37] allows users to manipulate images. BigGAN [34] and StyleGAN [35], [36] are powerful image composition models capable of generating a variety of high-quality images and have already been adopted by digital artists [38]. GAN models are rapidly developing and being widely used for text generation, image generation, video generation, and voice generation; however, the model structure varies considerably depending on the application field. The currently disclosed GAN models cannot be applied to the generation of 5G packet traffic without change and, therefore, a separate study is needed to generate 5G packet traffic with long-range dependency and bustiness characteristics.

COT-GAN [39] is an adversarial algorithm used to train implicit generative models optimized for the production of sequential data. The objective function of this algorithm is formulated using ideas from causal optimal transport (COT), which combines classic optimal transport methods with an additional temporal causality constraint. The COT-GAN algorithm is suitable for generating time series

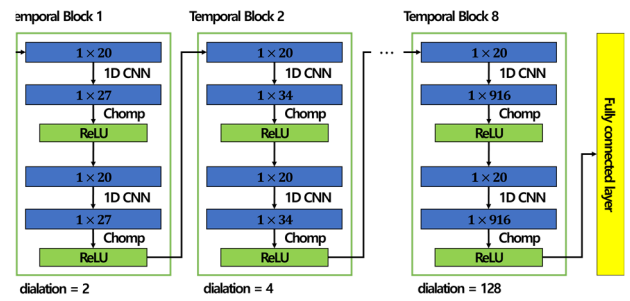


FIGURE 3. Discriminator network architecture

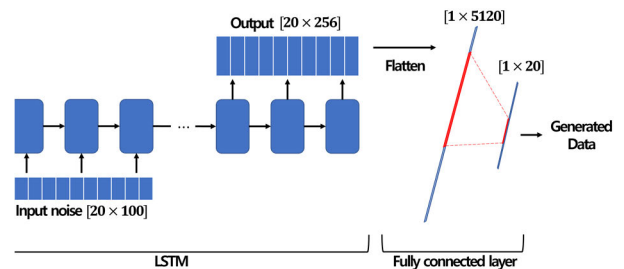


FIGURE 4. Generator network architecture

data when the real dataset consists of i.i.d. sequences or stationary time series. Although it opens the door to many applications that can benefit from time-series synthesis, the generation of ever-changing nonstationary 5G traffic requires separate research.

B. THE PROPOSED VIDEO TRAFFIC MODEL

We propose 5G Traffic-GAN, which achieves both higher computational efficiency and sample quality than autoregressive (AR) or stochastic models. The same as the standard GAN training method, the generative network generates 5G traffic candidates while the discriminative network evaluates them. The generator in the proposed GAN formulation transforms input noise $z \sim p_z$, a standard multivariate Gaussian, to the output $G(z)$ with distribution $p_g(x)$. The target data are sampled from an underlying distribution $p_d(x)$ and the discriminator $D(x)$ predicts the probability of its input coming from p_d . This is formulated as a two-player minimax game between the generator and the discriminator with the value function $V(G, D)$:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_d(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log (1 - D(G(z)))] \quad (1)$$

The generator and discriminator are the two players and take turns updating their model weights. In (1), the objective function of the discriminator is to maximize the likelihood of distinguishing between real data from the training set and fake data from the generator. Using cross-entropy to measure loss, $\log D(x)$, $D(x)$ is ideally 1 for labelled real training data x ; and for the fake data z generated from the generator G , its loss is $\log (1 - D(G(z)))$.

As shown in Fig. 2, the 5G Traffic-GAN has two parts. Both the generator and the discriminator are neural networks. The generator output is directly connected to the discriminator input through a hyperbolic tangent activation function. Through backpropagation, the discriminator's classification provides a signal that the generator uses to update its weights. The generator learns to generate plausible data so that the generated instances become negative training examples for the discriminator. The discriminator learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results. When training begins, the generator produces random data, and the discriminator quickly learns that it is not 5G traffic data. Finally, if the generator is well trained, the discriminator will be worse at distinguishing between the real trace and generated 5G traffic. As it starts to classify the generated data as real, its accuracy decreases. The discriminator uses the sigmoid activation function output in the range $[0, 1]$ to determine a value close to 1 as real and a value close to 0 as fake, but generator uses the hyperbolic tangent activation function to normalize the training data to a value in the range $[-1, 1]$. In the GAN generator, it is known that normalizing the training data to a value between -1 and 1 so that it has polarity can obtain stable model parameters.

As shown in Fig. 3, the discriminator consists of eight temporal blocks. Each temporal block has a structure in which 1×20 real traffic and generated traffic are alternately used as inputs, and the 1D CNN and chomp layers are repeated twice. The 1D CNN learns the characteristics of the input traffic, and the chomp layer makes the size of the output result of the 1D CNN the same as the initial input size so that it can be used as an input for the next 1D CNN. Stacking temporal blocks with dilated convolutions enables networks to have very large receptive fields with only eight layers, while preserving the input resolution throughout the network as well as computational efficiency. In this paper, the dilation is doubled for every layer up to a limit of 128. We use a fully connected layer and a sigmoid activation function in the output layer of the neural network to distinguish between the real trace and generated 5G traffic.

As shown in Fig. 4, the generator consists of LSTM networks that can remember past information. It generates traffic similar to real traffic by using the characteristics of remembering past information. A latent space sized 20×100 is used as the input of the LSTM and the number of LSTM hidden cells is 256. The output of the LSTM is serialized into a vector and used as the input of the fully connected layer, and a 1×20 traffic is generated. The generated traffic then passes through the hyperbolic tangent activation function. Latent space refers to an abstract multi-dimensional space containing feature values that cannot be interpreted directly, but which encodes a meaningful internal representation of externally observed events. In this paper, it is a 20×100 -dimensional hypersphere with each variable drawn from a Gaussian distribution with a mean of zero and a standard deviation of one. Through training, the generator

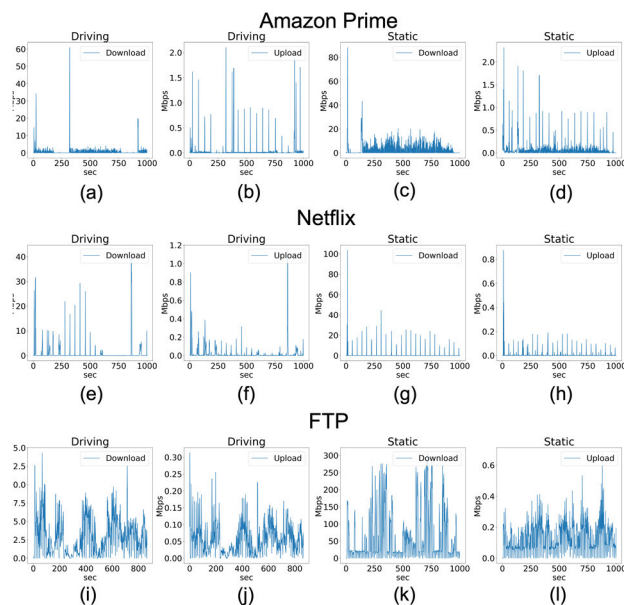


FIGURE 5. Data volume for each application of the dataset used for 5G Traffic-GAN training. (a) Amazon Prime downlink, driving. (b) Amazon Prime uplink, driving. (c) Amazon Prime downlink, static. (d) Amazon Prime uplink, static. (e) Netflix downlink, driving. (f) Netflix uplink, driving. (g) Netflix downlink, static. (h) Netflix uplink, static. (i) FTP downlink, driving. (j) FTP uplink, driving. (k) FTP downlink, static. (l) FTP uplink, static.

learns to map points into the latent space with specific output vectors and this mapping will be different each time the model is trained.

C. DATASET

The dataset used is a 5G trace dataset collected from a major Irish mobile operator [2]. It is generated from two mobility patterns (i.e., static and car) and across two application patterns (i.e., video streaming and file download). The video streaming dataset is a direct measurement of Netflix and Amazon Prime, which are representative OTT services, while watching on a mobile terminal. The dataset is composed of client-side cellular key performance indicators (KPIs) comprising channel-related metrics, context-related metrics, cell-related metrics, and throughput information. These metrics are generated from a well-known Android network monitoring application, G-NetTrack Pro. Fig. 5 shows the data volume over time for each application of the dataset in Kbps. Details of the training are described in Section IV.

D. OTHER TRAFFIC MODELS

The 5G Traffic-GAN is used to model the data traffic generated when streaming videos are downloaded from a server to a user equipment (UE). However, because the video streaming data transmitted from the camera mounted on the mobile robot to the controller is transmitted through the uplink, it is not appropriate to generate such uplink traffic using the 5G Traffic-GAN model. Thus, we generate uplink

video traffic using the existing stochastic video traffic model, the near real-time (NRT) video streaming model. In the NRT video streaming model, the packet sizes and packet inter-arrival time in a frame follow a truncated Pareto distribution. The parameter types and their specific values of the probability distribution are presented in [40] and [41]. As in [40], the inter-arrival time between the beginning of each frame is deterministic with 100 ms. The number of packets in a frame is also deterministic with eight packets per frame. The distribution of packet sizes is a truncated Pareto distribution with an average of 100 bytes. The probability density function (PDF) of the packet size for this model is as follows:

$$f_x = \begin{cases} \frac{\alpha k^\alpha}{x^{\alpha+1}}, & k \leq x < m \\ \left(\frac{k}{m}\right)^\alpha, & x = m \end{cases} \quad (2)$$

where α denotes the shape parameter known as the tail index, and k and m denote the minimum and maximum values in bytes, respectively. The inter-arrival time between packets has the same distribution as in (2) with different parameters.

Additionally, in this paper, a periodic traffic model and an aperiodic traffic model are used to simulate the traffic generated by the devices in the factory. The periodic traffic model is a simple model in which packets are continuously generated at regular intervals, given the packet length and transfer interval. The aperiodic traffic model generates packets with a given packet length with random inter-arrival times. In traffic simulation, this model is used differently to generate traffic for individual use cases in the simulation, as shown in Table 5 of Section VI.

E. 5G NR MAC FRAME SIMULATION

Because the 5G traffic generated by the proposed 5G TrafficGAN model is generated by learning the dataset directly collected by the 5G mobile terminal, the distribution of the generated packets is very similar to the actual traffic. However, because the traffic generated using a stochastic model such as a truncated Pareto distribution model simulates the packet generation pattern of the application layer or the network layer, the 5G new radio (NR) media access control (MAC) frame encapsulation process must be added to simulate the traffic shape during 5G transmission. In this paper, to convert IP packets generated by the probabilistic model into MAC frames, IP packets are collected as much as the MAC slot length (i.e., 0.5 ms) to create a 5G NR transport block (TB). The length of service data adaptation protocol (SDAP), packet data convergence protocol (PDCP), radio link control (RLC), and MAC headers are added to each TB to simulate 5G NR frame transmission.

IV. SPECTRUM CALCULATION RELATED WORKS

One of the main characteristics of 5G vertical services is the strict latency requirement. For example, the communication delay requirements between remote controllers for process

TABLE 1. Notations Related to Spectrum Calculation.

Notation	Description
$C(t)$	Time-varying system capacity [bps]
C	Maximum value of $C(t)$ [bps]
r_k, r_k'	Minimum data rate to transmit the k th frame within the delay constraint [bps]
t_k	Slot starting time of the k th frame [s]
V_k	Traffic volume of the k th frame [byte]
τ_k	Delay constraint [s]
T_s	MAC frame slot duration [s]
η	Spectral efficiency [bits/s/Hz]
F_{DL}	Spectrum requirement for DL [Hz]
F_{UL}	Spectrum requirement for UL [Hz]
F_{FDD}	Spectrum requirement for FDD system [Hz]
F_{TDD}	Spectrum requirement for TDD system [Hz]

automation in smart factories are very strict. On the other hand, file transfers using the file transfer protocol (FTP), such as log file transfers, have loose latency requirements. The system capacity to handle the same amount of traffic is larger when there is a strict delay requirement than when there is a loose delay requirement. The queuing theory can be used to calculate the system capacity considering the delay requirement; however, prior knowledge of the packet arrival process is required. A representative spectral requirement calculation recommendation to which the queuing theory is applied is [4]. In the absence of an appropriate stochastic traffic model, it is difficult to determine the system capacity using the queuing theory. In this paper, we propose a method for calculating the system capacity without stochastic model parameters of packet traffic whilst reflecting the delay constraints required by each application. By dividing the calculated system capacity by spectral efficiency, the required spectrum for the private 5G network is obtained.

A. SYSTEM CAPACITY CALCULATION

The notations used in Section IV are listed in Table 1. The basic concept of the proposed method is that the minimum required data rate satisfying the delay constraints is defined as the system capacity. The system capacity is time-varying because various services are processed in the system, and the required data transmission rate changes accordingly. The mathematical representation of the time-varying system capacity can be expressed as the sum of the minimum data transmission rates given in (3):

$$C(t) = \sum_k r_k [u(t - t_k) - u(t - t_k - \tau_k)], \quad (3)$$

where the unit step function $u(t)$ is defined as

$$u(t) = \begin{cases} 0, & x < 0 \\ 1, & x > 0. \end{cases} \quad (4)$$

The minimum data rate for transmitting the k th frame within the delay constraint τ_k is denoted by r_k , and the starting time of the k th frame is denoted as t_k . If the size of the k th frame is V_k bytes, the minimum data rate at which it is transmitted within τ_k s is calculated by dividing the traffic

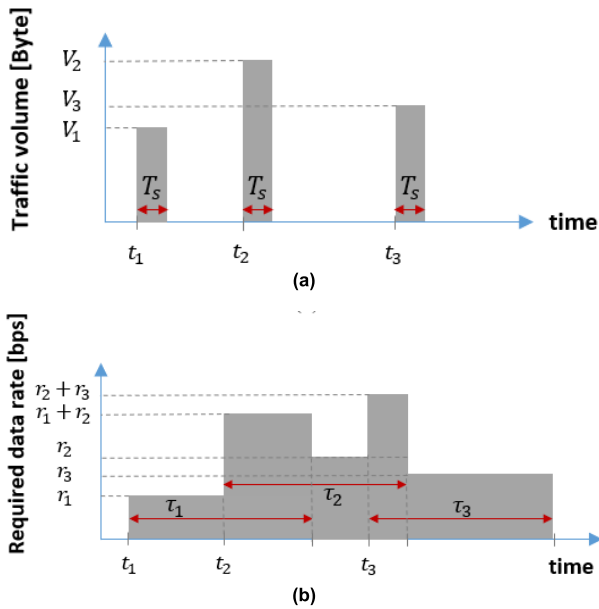


FIGURE 6. Example of FDD spectrum calculation process. (a) Generated 5G NR Frames. (b) Minimum data rates to meet each latency requirement.

volume by the latency as $r_k = 8V_k/\tau_k$ bps. For the system to satisfy all delay constraints, the system capacity should be selected as the maximum value of the time-varying system capacity given in (4), which is given as:

$$C = \max \{C(t)\}. \quad (5)$$

The duplexing type should be considered when calculating the spectrum requirements based on traffic volume. Frequency division duplexing (FDD) is implemented on a paired spectrum where downlink and uplink transmissions are sent on separate frequencies, whereas time division duplexing (TDD) is implemented on an unpaired spectrum, implying the usage of the same frequency for both downlink and uplink transmissions. Therefore, the spectrum requirement calculation is performed differently according to the duplexing type.

B. SPECTRUM REQUIREMENT: FDD

In FDD, the downlink and uplink frames are separated in the frequency domain and thus, the link capacity is calculated for each link. Assuming that in downlink, three MAC frames with V_1 , V_2 , and V_3 bytes have slot start times t_1 , t_2 , and t_3 as shown in Fig. 6(a), where T_s denotes the MAC frame slot duration; if each frame contains different service data that require different delay requirements τ_1 , τ_2 , and τ_3 , the minimum required data rates for transmitting each frame within the delay requirements are $r_1 = 8V_1/\tau_1$ bps, $r_2 = 8V_2/\tau_2$ bps, and $r_3 = 8V_3/\tau_3$ bps, respectively. For the link to transmit the corresponding MAC frame within the delay constraints, the system must maintain the minimum required data rate from the slot start time to the required delay value. Let us define this minimum required data rate as the system capacity considering the delay requirement. If $t_2 - t_1 < \tau_1$, the system

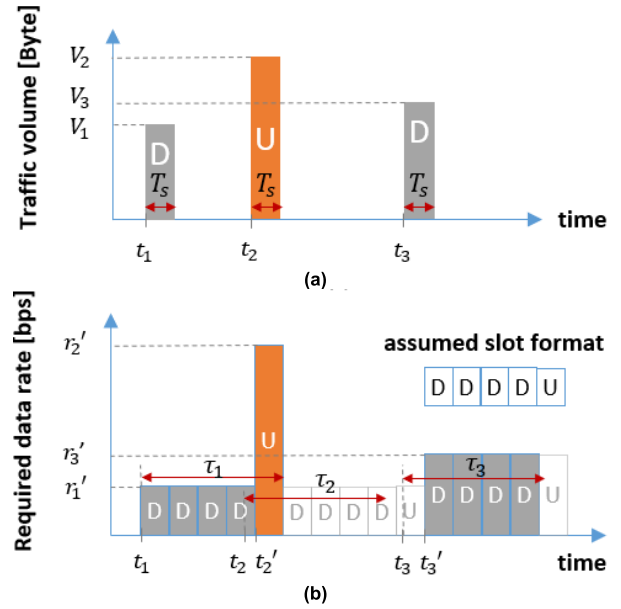


FIGURE 7. Example of TDD spectrum calculation process. (a) Generated 5G NR Frames. (b) Minimum data rates to meet each latency requirement.

transmits the second frame before the data transmission for the first frame is completed. In this case, the system capacity is $r_1 + r_2$, so that two frames can be transmitted within the delay requirements. This is depicted in Fig. 6(b). In Fig. 6(b), if $r_2 + r_3$ is the largest data rate value in the vertical axis, the system capacity is $r_2 + r_3$. The spectrum requirement of the downlink can be obtained by dividing the value by the spectral efficiency η as $F_{DL} = (r_2 + r_3)/\eta$. If the downlink and uplink spectrum requirements are F_{DL} and F_{UL} , respectively, the final spectrum requirement is given by (6):

$$F_{FDD} = 2 \times \max (F_{DL}, F_{UL}). \quad (6)$$

C. SPECTRUM REQUIREMENT: TDD

In TDD, as the downlink and uplink frames use the same frequency band, they are recognized in their time slots. The process of calculating the spectrum requirement in the TDD system is depicted in Fig. 7. Two downlink frames and one uplink frame are in a given time period, as shown in Fig. 7(a). Let the transmission time ratio of the downlink and uplink be 4:1; for convenience, the delay requirement of all frames is the same as $\tau = T_s/5$. As the frame starting at time t_1 needs to transmit data volume V_1 within the required delay τ , the minimum required transmission rate is $r_1' = 8V_1/(\tau \times 4/5)$, which is 5/4 times larger than that of FDD. The minimum required transmission rate $r_3' = 8V_3/(\tau \times 4/5)$ can be calculated in the same manner. In Fig. 7(b), if $r_3' > r_1'$, the downlink capacity is r_3' . The second frame is the uplink frame with the data volume V_2 should be transmitted within a slot allocated to the uplink in the required delay. This process is illustrated in Fig. 7(b). If r_2' is the maximum value among the minimum required data rates of the uplink frame, the uplink spectrum

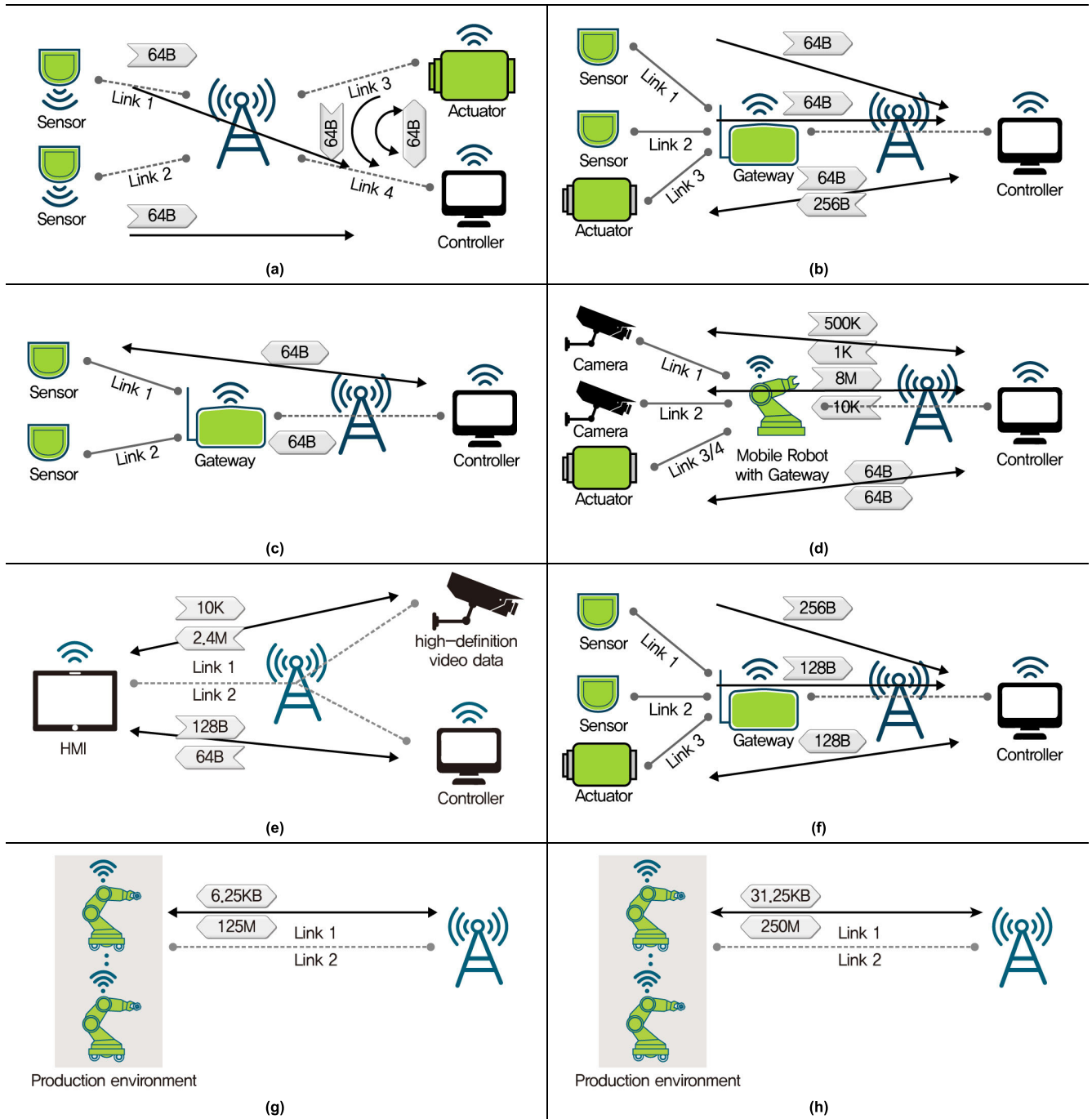


FIGURE 8. 5G industrial use cases. (a) Use Case 1: Motion Control, Mobile Controller. (b) Use Case 2: Closed Loop Control, Mobile I/O Gateway. (c) Use Case 3: Process Monitoring. (d) Use Case 4: Mobile Robot. (e) Use Case 5: Human-Machine Interface. (f) Use Case 6: Closed Loop Control for Process Automation. (g) Use Case 7: Control-to-Control (100 Mbps Link). (h) Use Case 8: Control-to-Control (1 Gbps Link).

requirement is a value obtained by dividing r'_2 by the spectrum efficiency. The downlink spectrum requirement is calculated for the downlink in the same manner. Finally, the spectrum requirement for the TDD system F_{TDD} is determined by (7):

$$F_{TDD} = \max(F_{DL}, F_{UL}). \quad (7)$$

V. USE CASE AND SCENARIO

5G wireless systems are expanding mobile communication services beyond mobile phones and broadband data services into new application areas, so-called vertical areas, including smart factories, smart cars, smart grids, and smart cities. Ho *et al.* [42] provided a survey of related research dedicated to automation in vertical domains. Reference [43] described

new use cases and potential requirements applicable to 5G systems for a 3GPP network operator to support 5G LAN-type services over the 5G system.

Applications in industrial automation systems have stringent requirements for latency and reliability. These requirements have already been met by current wired communication systems. To determine whether a wired system can be replaced by a wireless 5G system, it is necessary to analyze the actual data volumes and traffic types of industrial use cases. To accurately predict the network load, a realistic traffic model is required to enable the performance evaluation and design of the corresponding communications systems. 5G traffic models for industrial use cases are outlined in [44] and [45]. We performed simulations on eight use cases illustrated in Fig. 8. The descriptions of the individual use cases below summarize the white paper [45].

In Use Case 1, a controller with a wireless network connection communicates with two remote sensors (links 1 and 2) and a remote actuator (link 3). The two sensors send a small-sized message to the controller every 1 ms. The controller sends a command to the actuator every 10 ms and receives a response. It is assumed that they are transmitted as L2 frames over the reference interface with 64 bytes as the minimum size.

In Use Case 2, a mobile I/O gateway with a wireless network connection connects locally to two sensors and an actuator and communicates with a remote controller. The two sensors send a small-sized message to the remote controller every 1 ms through the I/O gateway. These messages are not acknowledged by the remote controller. The actuator receives a medium-sized message from the remote controller every 1 ms and acknowledges it with a small-sized message.

In Use Case 3, a mobile I/O gateway with radio network connectivity connects locally to two sensors and communicates with a remote computing entity such as a controller or a supervisory control and data acquisition (SCADA) system. The two sensors are polled by the remote computing entity and respond to that poll with their captured values; polling is performed every 20 ms on average.

In Use Case 4, a mobile I/O gateway with a wireless network connection mounted on a mobile robot has local connections to two high-definition cameras and two actuators. The camera on link 1 sends a continuous video stream to the remote controller at a data rate of 8 Mbps and receives its acknowledgment. The second camera only sends a video stream when the mobile robot reaches a defined location. This second camera has the same resolution and the same peak data rate as the first camera but has a lower average data rate as it only transfers images intermittently. Actuators communicating via link 3 receive a command from the remote controller every 10 ms and send a corresponding acknowledgment. The actuator using link 4 does the same every 1 ms.

In Use Case 5, mobile human-machine interface (HMI) devices with a wireless network connection have an emergency button system that communicates with the remote controller and sends a watchdog message every 5 ms that

the remote controller acknowledges. The HMI also receives a high-definition video stream.

In Use Case 6, a mobile I/O gateway with radio network connectivity connects locally to two groups of sensors and one group of actuators and communicates with a remote controller. The I/O gateway captures a value from each sensor in the first group and sends a message with the combined values of medium size every 200 ms and sends a smaller-sized message with the values from the other group every 500 ms via the I/O gateway to the remote controller. These messages are not acknowledged by the remote controller. The I/O gateway receives a message with the values from the group of actuators every 200 ms from the remote controller and sends back an acknowledgement message of the same size.

Use Cases 7 and 8 are control-to-control scenarios where two or more machines collaborate in modular production environments. Each machine communicates with every other machine, with similar traffic across all links. In traditional, non-flexible scenarios, 100 Mbps and 1 Gbps wired links are used (e.g., 1 Gbps links for video streaming and 100 Mbps links for motion control). In Use Case 7, for the 100 Mbps link, 50% periodic and 25% aperiodic traffic is assumed. Each machine sends and receives 6.25 KB of periodic data per 1 ms interval via link 1 from the collaborating machines. Aperiodic data can vary between 0 and 3.125 KB per 1 ms interval via link 2. In Use Case 8, for the 1 Gbps link, 25% periodic and 50% aperiodic traffic is assumed. Each machine sends and receives 31.25 KB of periodic data per 1 ms interval via link 1. Aperiodic data can vary between 0 and 62.5 KB per 1 ms interval via link 1.

Three deployment scenarios were considered in [45], namely small-scale, large-scale deployment, and inbound logistics. Among the three examples, small-scale deployment and large-scale deployment are considered in this study. Tables 2 and 3 present the two deployment scenarios and the corresponding 5G traffic.

TABLE 2. Scenario – Small scale deployment scenario.

Use Case	Number of links	Number of devices	Latency [ms]	Availability [%]
Use Case 1	4	150	2	99.9
Use Case 2, 6	4	15	10	99.9
Use Case 3	4	20	300	99.9
Use Case 4	4	8	50	99.9
Use Case 5	4	7	10	99.9
Use Case 7	4	0	10	99.9

VI. IMPLEMENTATION AND SIMULATION STUDY

A. 5G TRAFFIC-GAN RESULTS

The dataset [2] is a collection of 216,000 s (i.e., 60 h) of video streaming traffic generated while watching representative OTT services (e.g., Netflix and Amazon Prime) using the Android application, G-NetTrack Pro. Only data measured in the stationary state were used for model training, and

TABLE 3. Scenario – Large scale deployment scenario.

Use Case	Number of links	Number of devices	Latency [ms]	Availability [%]
Use Case 1	4	300	2	99.9
Use Case 2, 6	4	30	10	99.9
Use Case 3	4	40	300	99.9
Use Case 4	4	40	50	99.9
Use Case 5	4	14	10	99.9
Use Case 7	4	2	10	99.9

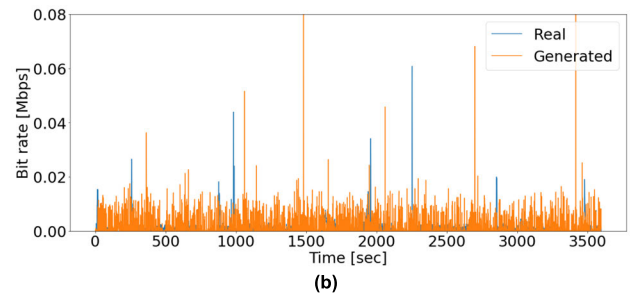
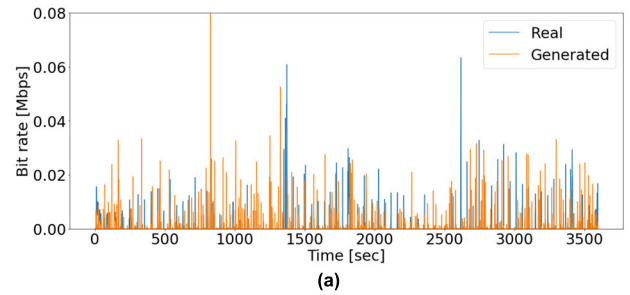
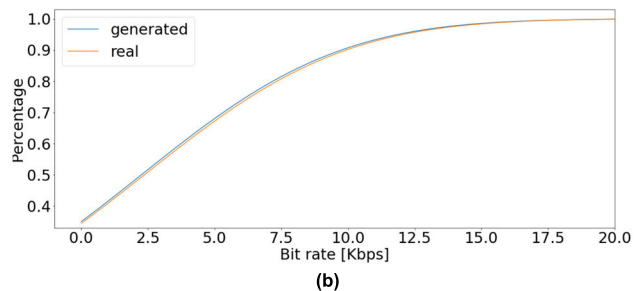
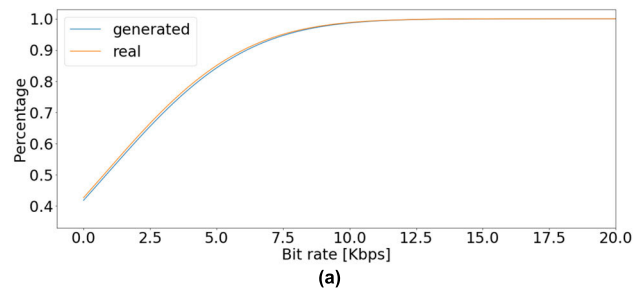
only Timestamp, UL_bitrate, and DL_bitrate among more than 20 fields were used as inputs to the model. Given the dataset [2], Netflix has a longer data request cycle than Amazon Prime and requests a large number of video chunks at a time. As shown in Fig. 1, there is a clear difference in the pattern of downlink traffic. Unlike Netflix, Amazon Prime tends to constantly generate small amounts of data.

Of the 60-hour dataset, 58 hours were used for training and validation, and the remaining two hours were used for testing. The hyperparameters used to train the model are as follows: In the generator model, Adam was used as an optimizer to learn 4098 parameters of the generator network, and the learning rate was set to 0.01. Dropout is not used, and the number of hidden LSTM cells is 256. In the discriminator model, the same optimizer and learning rate as the generator were used to learn 502 parameters of the discriminator network, and dropout was not applied. The total number of time blocks is 8, and the input data were trained using the extended causal convolution technique [46].

The computing environment in which 5G Traffic-GAN learned the dataset of [2] was a personal computer equipped with an AMD Ryzen 7-1700 8-core CPU, 16GB RAM, and NVIDIA GTX 1070ti 6GB GPU, which took 565 s to learn the entire dataset once. The 5G Traffic-GAN was trained for a total of 650 epochs, so the total time spent on training corresponds to 102 hours ($565 \text{ s} \times 650 = 367,250 \text{ s}$). In the same environment, when the 5G Traffic-GAN generates traffic, it varies depending on the scenario, but in most cases, an inference time of 60 s was required. In other words, traffic begins to be generated after 60 s, and in this experiment, traffic simulation was performed for 15 minutes to obtain sufficient traffic.

Fig. 9 shows the pattern of traffic volume over time generated by the 5G Traffic-GAN overlaid with the actual traffic in the dataset. It can be seen that both the generated Netflix shown in Fig. 9(a) and Amazon Prime shown in Fig. 9(b) traffics are generated similarly to the actual traces with the occurrence cycle and pattern.

To compare the generated video streaming traffic with the traffic given in the dataset, a cumulative distribution function (CDF) was observed. Fig. 10 shows an overlay of the CDF to visualize how similar the generated traffic is to real traffic. In particular, it shows data between 0-20 Kbps, where there is a significant change in the rate of traffic. It can be seen that both Netflix in Fig. 10(a) and Amazon Prime in Fig. 10(b) have very similar CDFs of generated traffic and actual traffic.

**FIGURE 9. An overlapping representation of the generated downlink packet traffic and real packet traffic. (a) Netflix. (b) Amazon Prime.****FIGURE 10. CDF comparison of generated and real downlink packet traffic. (a) Netflix. (b) Amazon Prime.**

To evaluate the performance of the proposed traffic generation model more accurately, we observed Jensen-Shannon divergence (JSD) and maximum mean discrepancy (MMD). Jensen-Shannon divergence is a method of measuring the similarity between two probability distributions P and Q . Here, P is the probability distribution generated by the proposed model, and Q is the probability distribution of the actual dataset. The Jensen-Shannon divergence is a symmetrized and smoothed version of the Kullback–Leibler (KL) divergence $D(P||Q)$. It is defined by

$$JSD(P||Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M), \quad (8)$$

where $M = 0.5(P+Q)$. The maximum mean discrepancy is a kernel-based statistical test used to determine whether the two distributions P and Q are the same. MMD is defined as [47]

$$\begin{aligned} &MMD(P, Q) \\ &= \left[\mathbb{E}_{x,x'}(k(x, x')) + \mathbb{E}_{y,y'}(k(y, y')) - 2\mathbb{E}_{x,y}(k(x, y)) \right]^{\frac{1}{2}}, \end{aligned} \quad (9)$$

where x and x' are independent random variable drawn according to probability distribution P, y and y' are independent random variables drawn according to the distribution Q, and x is independent of y .

We evaluated the performance by changing the structure of the generator and discriminator models. Table 4 shows the applied models and the performance evaluation results. The first column of Table 4 shows the generator and discriminator structures of the proposed 5G Traffic-GAN model. For example, in the LSTM-TCN model in the third row, the generator is the LSTM structure, and the discriminator is the TCN structure.

TABLE 4. Comparison of performance of ML models.

Model	Jensen-Shannon Divergence	Maximum Mean Discrepancy
TCN-TCN (Amazon)	0.006753	0.123564
TCN-TCN (Netflix)	0.015946	0.164307
LSTM-TCN (Amazon)	0.000499	0.067060
LSTM-TCN (Netflix)	0.000163	0.030687
2LSTM-TCN (Amazon)	0.006547	0.028672
2LSTM-TCN (Netflix)	0.002708	0.012222
WGAN-GP (Amazon)	0.000086	0.169899
WGAN-GP (Netflix)	0.000046	0.015154

The first model, TCN-TCN is the case where both the generator and the discriminator have a TCN structure, the second model, LSTM-TCN is the case where the discriminator is the same as the first model and the LSTM structure is applied to the generator. The third model, 2LSTM-TCN is the case where the LSTM layer is doubled in the second model. The fourth model, WGAN-GP is a case of generating traffic using Wasserstein GAN-gradient penalty (WGAN-GP) model. As shown in Table 4, the model applying LSTM to the generator model showed small values of Jensen-Shannon divergence and maximum mean discrepancy, so it seems that the time series features and patterns of the given measured data were effectively trained. The model composed of both the generator and discriminator with TCN generated similar data patterns, but failed to properly express time series characteristics, which were measured with higher Jensen-Shannon divergence and a maximum mean discrepancy than LSTM-TCN, as shown in Table 4. When the LSTM layer was doubled, the time series characteristics were well expressed, but the data distribution was not accurately learned. Overall, the WGAN-GP model showed a relatively lower performance than the model using TCN and LSTM.

B. SPECTRUM CALCULATION RESULTS

The proposed 5G spectrum calculator is implemented based on Python 3.8.2 and provides a user-friendly interface. It includes a traffic generation function that can simulate various 5G traffic such as periodic packet data, aperiodic packet data, stochastic traffic (e.g., NRT video streaming), and traffic that mimics real traces. The spectrum calculator developed in this study allows users to choose the deployment scenarios given in Tables 2 and 3 and set the parameters for each of the eight use cases presented in Table 5. Although the user cannot set it in the user interface, there are MAC frame durations and spectral efficiencies, which are important parameters for calculating spectrum requirements. In this simulation, the MAC frame duration was set to 0.5 ms, and the spectral efficiency was 13.9 b/s/Hz for downlink and 7.7 b/s/Hz for uplink [48]. After completing the parameter setting and pressing the Run button, 5G traffic is generated for the scenario, and the amount of spectrum that can accommodate the generated 5G traffic is estimated. As shown in Fig. 11, the user inputs the following information:

- Choose a “Scenario”. There are four selectable scenarios, which are “Small scale deployment scenario”, “Large scale deployment scenario”, “Inbound logistics deployment scenario”, and “User-created scenario”.
- Select duplexing type. Users can choose either a TDD or FDD. If the TDD is selected, the ratio of the downlink and uplink in the time axis must also be entered.
- Enter traffic simulation-related parameters for each use case in the Attributes pane. The parameters related to the traffic simulation to be entered are the number of links, number of devices, and latency requirements.
- You can create traffic suitable for the scenario by clicking the Run button in the upper right corner. Traffic is generated using the parameters listed in Table 5.

After the traffic simulation, the uplink and downlink traffic demand appear in the “Results” window on the right, and the spectrum requirement is calculated and displayed in the lower window of Fig. 11. The traffic simulation results generated for each use case can be viewed in separate window. Fig. 12 shows a portion of the generated traffic for each use case.

In the experiment, use cases applied to the factory layout given in [45] were assumed. The layout is a typical factory for discrete production and assembly. It comprises a production area, assembly areas, a warehouse, a commissioning space, and office cubicles, spanning a total of approximately 15,000 square meters with a ceiling 30 m high. A large-scale deployment scenario in which sensors and other 5G devices are densely placed in the space and a small-scale deployment scenario in which the density is low were simulated.

First, the experimental results for the large-scale deployment scenario are examined. As shown in Fig. 11, traffic simulations for large deployment scenario, including six use cases, resulted in a total of 625.7 Mb/s on the uplink and an average of 741.4 Mb/s on downlink. This is the sum of all the traffic from the six use cases that constitute the scenario.

TABLE 5. Traffic simulation parameters of eight use cases.

Use Case #	Traffic Model	Traffic Parameters	Link 1		Link 2		Link 3		Link 4	
			UL	DL	UL	DL	UL	DL	UL	DL
1	Periodic	Transfer Interval [ms]		1		1		1	10	10
		Message Size [Bytes]		64		64		64	64	64
2	Periodic	Transfer Interval [ms]	1		1		1	1		
		Message Size [Bytes]	64		64		64	256		
3	Periodic	Transfer Interval [ms]	20	20	20	20				
		Message Size [Bytes]	64	64	64	64				
4	Periodic	Transfer Interval [ms]					10	10	1	1
		Message Size [Bytes]					64	64	64	64
	NRT Video (UL), Aperiodic (DL)	Average Data Rate [Kbps]	8,000	10	500	1				
		Peak Data Rate [Kbps]	8,000	10	4,000	10				
5	Periodic	Transfer Interval [ms]	5	5						
		Message Size [Bytes]	128	64						
	Aperiodic (UL), 5G Traffic-GAN (DL)	Average Data Rate [Kbps]			10	2,400				
		Peak Data Rate [Kbps]			10	14,000				
6	Periodic	Transfer Interval [ms]	200		500		200	200		
		Message Size [Bytes]	256		128		128	128		
7	Periodic	Transfer Interval [ms]	1	1						
		Message Size [Bytes]	6,250	6,250						
	Aperiodic	Average Data Rate [Kbps]			12,500	12,500				
		Peak Data Rate [Kbps]			25,000	25,000				
8	Periodic	Transfer Interval [ms]	1	1						
		Message Size [Bytes]	31,250	31,250						
	Aperiodic	Average Data Rate [Kbps]			250,000	250,000				
		Peak Data Rate [Kbps]			500,000	500,000				

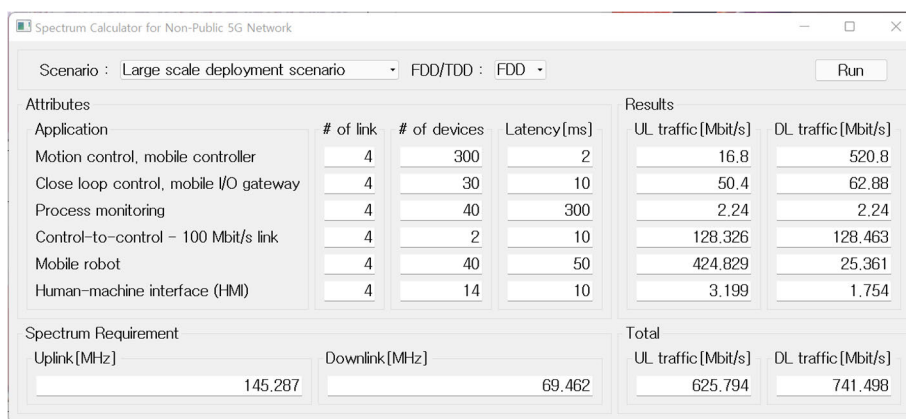


FIGURE 11. Parameter setting and simulation results for each use case. This is the result of selecting a large-scale deployment scenario and FDD.

When using FDD duplex, the spectrum was calculated in consideration of packet traffic delay requirements. It was calculated that a spectrum of 145.2 MHz was required for uplink and 69.4 MHz for downlink.

When TDD was selected, a large-scale deployment scenario traffic simulation with six use cases resulted in an average of 625.4 Mb/s traffic generated on uplink and 741.3 Mb/s traffic generated on downlink. This is very similar to the results shown in the sum window at the bottom right of Fig. 11. Similar traffic loads are generated because they are generated using the same parameters. When selecting TDD duplex, the DL:UL ratio should be set. The default value is 4:1, but this can be changed. In the simulation of this paper, the default value was used. Through the simulation, the

spectrum was calculated in consideration of packet traffic delay requirements. It was calculated that a spectrum of 397.8 MHz was required for uplink and 70.5 MHz for downlink. In this paper, the traffic simulation and spectrum requirement calculation results of the TDD large-scale deployment scenario are not shown in a figure. The reason that the uplink spectrum requirement is calculated to be large despite the small traffic volume is because the DL:UL ratio is 4:1. Since the uplink is allocated a shorter time than the downlink, a wide spectrum is required to process the generated traffic within given delay requirements.

Next, the experimental results for the small-scale deployment scenario are examined. As a result of performing a small-scale deployment scenario simulation including six use

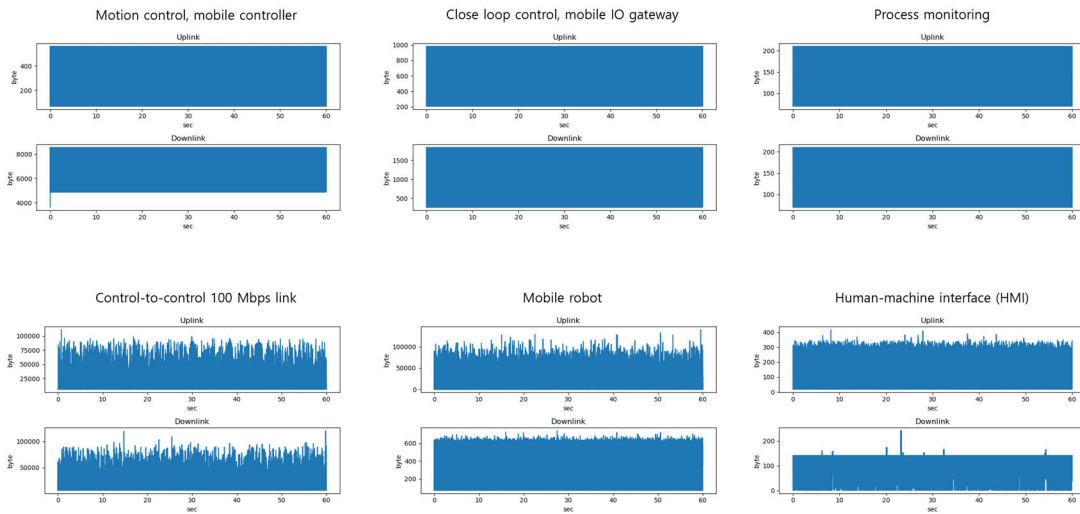


FIGURE 12. Traffic simulation results for each use case. This is the result of selecting a large-scale deployment scenario.

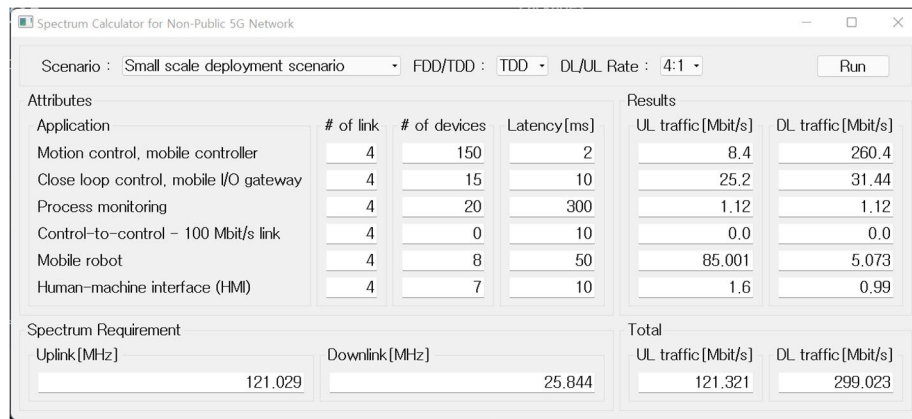


FIGURE 13. Parameter setting and simulation results for each use case. This is the result of selecting a small-scale deployment scenario and TDD.

cases, an average of 121.2 Mb/s of traffic was generated in the uplink and an average of 298.9 Mb/s of traffic was generated in the downlink. In case of using FDD duplex, as a result of calculating the spectrum requirements considering the delay of packet traffic generated through traffic simulation, it was calculated that 31.0 MHz of spectrum is required for uplink and 22.0 MHz of spectrum is required for downlink. In the case of the small-scale deployment scenario, since the amount of traffic generated is smaller than that in the large-scale deployment scenario, it can be seen that the amount of spectrum required is also small.

When TDD is used, the traffic parameters used for each use case of the small-scale deployment scenario and the calculated spectrum requirements are shown in Fig. 13. Also, the traffic simulation results for each use case are shown in Fig. 14. When selecting TDD, the DL:UL ratio should be set. The default value is 4:1, but this can be changed. In the simulation of this paper, the default value was used.

As shown in Fig. 13, traffic simulations for the small-scale deployment scenario, including six use cases, resulted in a total of 121.3 Mb/s on uplink and an average of 299.0 Mb/s on the downlink. This is the sum of all the traffic from the six use cases that constitute the scenario. This is the sum of all the traffic from the six use cases that make up the scenario. In the case of using the TDD duplex, because of calculating the spectrum requirements considering the delay of packet traffic generated through traffic simulation, it was calculated that 121.0 MHz of spectrum is required for uplink and 25.8 MHz of spectrum is required for downlink. The generated traffic volume and spectrum requirements are shown at the bottom of Fig. 13. The reason that the uplink spectrum requirement is calculated to be large despite the small traffic volume is because the DL:UL ratio is 4:1. Since the uplink is allocated a shorter time than the downlink, a wide spectrum is required to process the generated traffic within given delay requirements.

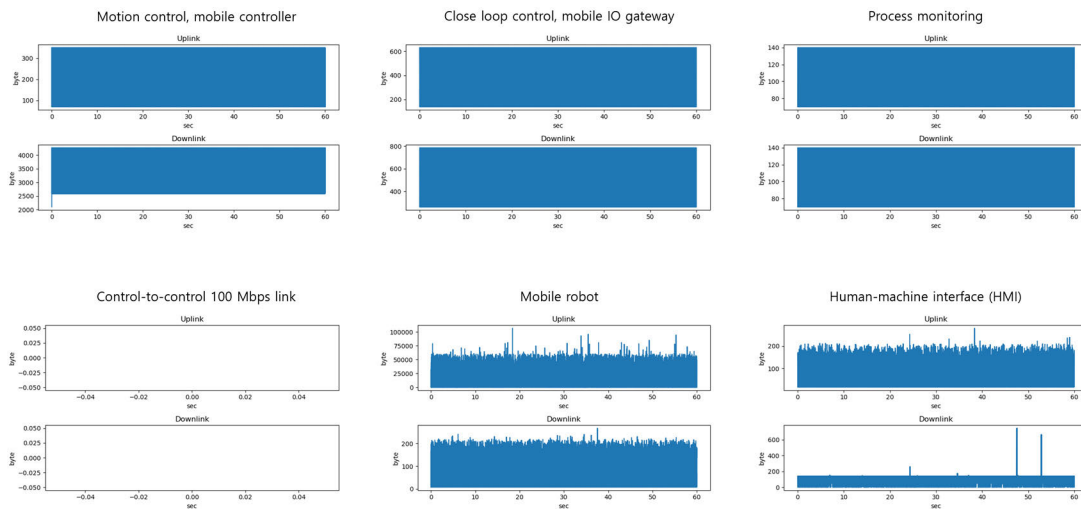


FIGURE 14. Traffic simulation results for each use case. This is the result of selecting a small-scale deployment scenario.

In addition, inbound logistics deployment scenarios and user-created scenarios can be configured for both TDD and FDD, but the simulation results are not included in this paper.

VII. CONCLUSION

In this paper, we proposed a neural network-based 5G traffic generation model and a methodology for calculating the necessary spectrum requirements of private 5G networks. To accurately estimate the spectral requirements, an analysis of the actual data volume and traffic type of the place where the network is to be built is necessary. However, there is currently no suitable traffic generation model to test the load on a private 5G network, we developed a GAN-based traffic generation model that can generate realistic traffic by learning the real traffic traces collected from a major mobile network operator. In the case of industrial applications, probability-based traffic models were also used in parallel because there were not enough datasets to learn. To estimate the spectrum requirements, the proposed 5G traffic generation model was combined with the proposed 5G spectrum calculator. In this paper, spectrum calculation was performed differently according to the two duplexing types, FDD and TDD. As a guide for companies that implement actual industrial use cases with 5G networks, we simulated eight use cases defined in the 5G ACIA white paper. Spectrum requirements were observed while changing traffic loads, deployment scenarios, and duplexing schemes. Various experiments have confirmed that a bandwidth of at least 22.0 MHz to a maximum of 397.8 MHz is required depending on the deployment scenario.

ACKNOWLEDGMENT

The experiments in this paper were performed using GPU resources provided by the Korea National IT Industry Promotion Agency (NIPA). In addition, GPU resources purchased

through the following research project were partially used. The research project is the ‘‘Convergence and Open Sharing System’’ Project, supported by the Ministry of Education and National Research Foundation of Korea.

REFERENCES

- [1] A. Thantharate. (2019). *5G, Network Slicing, 5G Security & Deep Learning*. [Online]. Available: <https://github.com/adtmv7/DeepSlice>
- [2] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, ‘‘Beyond throughput, the next generation: A 5G dataset with channel and context metrics,’’ in *Proc. 11th ACM Multimedia Syst. Conf.*, Istanbul, Turkey, May 2020, pp. 303–308. [Online]. Available: <https://github.com/uccmis1/5Gdataset>
- [3] *Methodology for the Calculation of IMT-2000 Terrestrial Spectrum Requirements*, document Rec. ITU-R Rec. M.1390, ITU, 1999.
- [4] *Methodology for Calculation of Spectrum Requirements for the Terrestrial Component of International Mobile Telecommunications*, document Rec. ITU-R Rec. M.1768-1, ITU, 2013.
- [5] H.-G. Shin, H.-J. Kim, S.-W. Lee, H.-G. Yoon, and Y.-H. Choi, ‘‘Calculation of AeroMACS spectrum requirements based on traffic simulator,’’ *Sensors*, vol. 21, no. 10, p. 3343, May 2021, doi: [10.3390/s21103343](https://doi.org/10.3390/s21103343).
- [6] Y. Kim and S. Park, ‘‘Calculation method of spectrum requirement for IMT-2020 eMBB and URLLC with puncturing based on M/G/1 priority queuing model,’’ *IEEE Access*, vol. 8, pp. 25027–25040, 2020, doi: [10.1109/ACCESS.2020.2971223](https://doi.org/10.1109/ACCESS.2020.2971223).
- [7] P. T. Brady, ‘‘A model for generating on-off speech patterns in two-way conversation,’’ *Bell Syst. Tech. J.*, vol. 48, no. 7, pp. 2445–2472, Sep. 1969.
- [8] J. N. Daigle and J. D. Langford, ‘‘Models for analysis of packet voice communications systems,’’ *IEEE J. Sel. Areas Commun.*, vol. JSAC-4, no. 6, pp. 847–855, Sep. 1986, doi: [10.1109/JSAC.1986.1146403](https://doi.org/10.1109/JSAC.1986.1146403).
- [9] D. LeGall, ‘‘MPEG: A video compression standard for multimedia applications,’’ *Commun. ACM*, vol. 34, no. 4, pp. 305–313, 1991.
- [10] T. Chiang and D. Anastassiou, ‘‘Hierarchical coding of digital television,’’ *IEEE Commun. Mag.*, vol. 32, no. 5, pp. 38–45, May 1994, doi: [10.1109/35.281577](https://doi.org/10.1109/35.281577).
- [11] P. Pancha and M. El Zarki, ‘‘MPEG coding for variable bit rate video transmission,’’ *IEEE Commun. Mag.*, vol. 32, no. 5, pp. 54–66, May 1994, doi: [10.1109/35.281579](https://doi.org/10.1109/35.281579).
- [12] M. W. Garrett and W. Willinger, ‘‘Analysis, modeling and generation of self-similar VBR video traffic,’’ in *Proc. Conf. Commun. Archit., Protocols Appl. SIGCOMM*, London, U.K., 1994, pp. 269–280.
- [13] S. Tanwir, D. Nayak, and H. Perros, ‘‘Modeling 3D video traffic using a Markov modulated gamma process,’’ in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2016, pp. 1–6, doi: [10.1109/ICNC.2016.7440638](https://doi.org/10.1109/ICNC.2016.7440638).

- [14] J. Zhang, W. Ouyang, X. Zhang, D. Wang, and J. Tang, "A survey of network traffic generation," in *Proc. 3rd Int. Conf. Cyberspace Technol. (CCT)*, 2015, pp. 1–6, doi: [10.1049/cp.2015.0862](https://doi.org/10.1049/cp.2015.0862).
- [15] B. Chandrasekaran, "Survey of network traffic models," Dept. Comput. Sci. Eng., Washington Univ. St. Louis, St. Louis, MO, USA, 2009, vol. 567. [Online]. Available: [http://www.cse.wustl.edu/~jain/cse567-06/traffic models3.htm](http://www.cse.wustl.edu/~jain/cse567-06/traffic%20models3.htm)
- [16] S. Tanwir and H. Perros, "A survey of VBR video traffic models," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1778–1802, 4th Quart., 2013, doi: [10.1109/SURV.2013.010413.00071](https://doi.org/10.1109/SURV.2013.010413.00071).
- [17] G. F. Riley and T. R. Henderson, "The ns – 3 network simulator," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Günes, and J. Gross, Eds. Berlin, Germany: Springer, 2010, pp. 15–34.
- [18] Netropy Traffic Generators. (2021). *Apposite Technology*. [Online]. Available: <https://www.apposite-tech.com/>
- [19] H.-W. Kim, J.-H. Lee, Y.-H. Choi, Y.-U. Chung, and H. Lee, "Dynamic bandwidth provisioning using ARIMA-based traffic forecasting for mobile WiMAX," *Comput. Commun.*, vol. 34, no. 1, pp. 99–106, Jan. 2011, doi: [10.1016/j.comcom.2010.08.008](https://doi.org/10.1016/j.comcom.2010.08.008).
- [20] *The National Laboratory for Applied Network Research (NLNAR) Project*. Accessed: 2011. [Online]. Available: <http://www.nlanr.net>
- [21] T. Otsoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi, and K. Shiimoto, "Traffic prediction for dynamic traffic engineering," *Comput. Netw.*, vol. 85, pp. 36–50, Jul. 2015, doi: [10.1016/j.comnet.2015.05.001](https://doi.org/10.1016/j.comnet.2015.05.001).
- [22] J. Guo, Y. Peng, X. Peng, Q. Chen, J. Yu, and Y. Dai, "Traffic forecasting for mobile networks with multiplicative seasonal ARIMA models," in *Proc. 9th Int. Conf. Electron. Meas. Instrum.*, Aug. 2009, pp. 3/377–3/380, doi: [10.1109/ICEMI.2009.5274287](https://doi.org/10.1109/ICEMI.2009.5274287).
- [23] C. Chen, Q. Pei, and L. Ning, "Forecasting 802.11 traffic using seasonal ARIMA model," in *Proc. Int. Forum Comput. Sci.-Technol. Appl.*, 2009, pp. 347–350, doi: [10.1109/IFCSTA.2009.207](https://doi.org/10.1109/IFCSTA.2009.207).
- [24] J. Fan, D. Mu, and Y. Liu, "Research on network traffic prediction model based on neural network," in *Proc. 2nd Int. Conf. Inf. Syst. Comput. Aided Educ. (ICISCAE)*, Sep. 2019, pp. 554–557, doi: [10.1109/ICIS-CAE48440.2019.221694](https://doi.org/10.1109/ICIS-CAE48440.2019.221694).
- [25] X. Xu, S. Gao, and Z. Jiang, "LSTCN: An attention-based deep neural network model combining LSTM and TCN for cellular network traffic prediction," in *Proc. 5th Int. Conf. Commun. Inf. Syst. (ICISIS)*, Oct. 2021, pp. 34–38, doi: [10.1109/ICISIS53528.2021.9645961](https://doi.org/10.1109/ICISIS53528.2021.9645961).
- [26] N. Piovesan, A. De Domenico, M. Bernabe, D. Lopez-Perez, H. Baohongqiang, G. Xinli, W. Xie, and M. Debbah, "Forecasting mobile traffic to achieve greener 5G networks: When machine learning is key," in *Proc. IEEE 22nd Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sep. 2021, pp. 276–280, doi: [10.1109/SPAWC51858.2021.9593102](https://doi.org/10.1109/SPAWC51858.2021.9593102).
- [27] A. Cheng, "PAC-GAN: Packet generation of network traffic using generative adversarial networks," in *Proc. IEEE 10th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Oct. 2019, pp. 0728–0734, doi: [10.1109/IEMCON.2019.8936224](https://doi.org/10.1109/IEMCON.2019.8936224).
- [28] M. Rigaki and S. Garcia, "Bringing a GAN to a knife-fight: Adapting malware communication to avoid detection," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 70–75, doi: [10.1109/SPW.2018.00019](https://doi.org/10.1109/SPW.2018.00019).
- [29] M. Ring, D. Schlör, D. Landes, and A. Hotho, "Flow-based network traffic generation using generative adversarial networks," 2018, *arXiv:1810.07795*.
- [30] M. R. Shahid, G. Blanc, H. Jmila, Z. Zhang, and H. Debar, "Generative deep learning for Internet of Things network traffic generation," in *Proc. IEEE 25th Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Dec. 2020, pp. 70–79, doi: [10.1109/PRDC50213.2020.00018](https://doi.org/10.1109/PRDC50213.2020.00018).
- [31] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, "DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks," in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2019, pp. 0762–0767, doi: [10.1109/UEMCON47517.2019.8993066](https://doi.org/10.1109/UEMCON47517.2019.8993066).
- [32] A. Thantharate, R. Paropkari, V. Walunj, C. Beard, and P. Kankariya, "Secure5G: A deep learning framework towards a secure network slicing in 5G and beyond," in *Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2020, pp. 0852–0857, doi: [10.1109/CCWC47524.2020.9031158](https://doi.org/10.1109/CCWC47524.2020.9031158).
- [33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.
- [34] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. ICLR*, Sep. 2019, pp. 1–39.
- [35] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.
- [36] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.
- [37] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
- [38] J. Bailey, *The Tools of Generative Art, From Flash to Neural Networks*. New York, NY, USA: Art America, Jan. 2020.
- [39] T. Xu, K. Li Wenliang, M. Munn, and B. Acciaio, "COT-GAN: Generating sequential data via causal optimal transport," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 8798–8809.
- [40] R. Srinivasan, *IEEE 802.16m Evaluation Methodology Document (EMD)*, document IEEE 802.16m-08/004r2, Jul. 2008.
- [41] *cdma2000 Evaluation Methodology Revision A*, document Rec. C.R1002-A, 3GPP2, May 2009.
- [42] T. M. Ho, T. D. Tran, T. T. Nguyen, S. M. A. Kazmi, L. B. Le, C. S. Hong, and L. Hanzo, "Next-generation wireless solutions for the smart factory, smart vehicles, the smart grid and smart cities," 2019, *arXiv:1907.10102*.
- [43] *Technical Specification Group Services and System Aspects; Feasibility Study on LAN Support in 5G (Release 16)*, document Rec. TR 22.821, 3GPP, Jun. 2018.
- [44] *Technical Specification Group Services and System Aspects; Service Requirements for Cyber-Physical Control Applications in Vertical Domains; Stage 1 (Release 18)*, document Rec. TS 22.104, Sep. 2021.
- [45] 5G-ACIA and ZVEI, Frankfurt, Germany. (2019). *A 5G Traffic Model for Industrial Use Cases*. [Online]. Available: https://5g-acia.org/wp-content/uploads/2021/04/WP_5G_5G_Traffic_Model_for_Industrial_UseCases_22.10.19.pdf
- [46] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.
- [47] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, 2012.
- [48] Y. Kim, F. Sun, Y. Wang, Y. Qi, J. Lee, Y. Kim, J. Oh, H. Ji, J. Yeo, S. Choi, H. Ryu, H. Noh, and T. Kim, "New radio (NR) and its evolution toward 5G-advanced," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 2–7, Jun. 2019, doi: [10.1109/MWC.2019.8752473](https://doi.org/10.1109/MWC.2019.8752473).



DAEGYEOM KIM received the A.S. degree in computer and mobile convergence engineering from the Gyeonggi University of Science and Technology, Gyeonggi-do, South Korea, in 2021, and the B.S. degree in computer engineering from the Korean Academic Credit Bank System, Seoul, South Korea, in 2021. He is currently pursuing the M.S. degree in robotics with Kwangwoon University, Seoul. His research interests include communication processing, speech recognition and synthesis, and stock price forecasting using natural language processing.



MYEONGJIN KO received the A.S. degree in computer and mobile convergence engineering from the Gyeonggi University of Science and Technology, Gyeonggi-do, South Korea, in 2021, and the B.S. degree in computer engineering from the Korean Academic Credit Bank System, Seoul, South Korea, in 2021. He is currently pursuing the M.S. degree in robotics with Kwangwoon University, Seoul.

His research interests include communication networks, software development, natural language processing, singing voice synthesis, and speech recognition and synthesis.

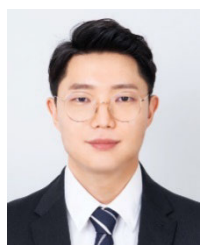


positioning technology, machine learning, and speech recognition and synthesis

SUNGHYUN KIM received the A.S. degree in computer and mobile convergence engineering from the Gyeonggi University of Science and Technology, Gyeonggi-do, South Korea, in 2020, and the B.S. degree in computer engineering from the Korean Academic Credit Bank System, Seoul, South Korea, in 2020. He is currently pursuing the M.S. degree in robotics with Kwangwoon University, Seoul. His research interests include communication networks, software development, machine learning, and speech recognition and synthesis



YUNBAE KIM received the B.Sc. and Ph.D. degrees in mathematical sciences from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2009 and 2015, respectively. Since November 2015, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon. His research interests include performance evaluation and optimization of wireless communication networks.



SUNGWOO MOON received the B.S. degree in robotics from Kwangwoon University, Seoul, South Korea, in 2020, where he is currently pursuing the M.S. degree in robotics. His research interests include communication networks, software development, positioning technology, machine learning, deep learning, and speech recognition and synthesis.



HYUNGOO YOON received the B.S., M.S., and Ph.D. degrees in electronics engineering from Yonsei University, Seoul, South Korea, in 1995, 1997, and 2002, respectively. From 2002 to 2004, he was with Hyundai Electronics, Icheon, where he had developed code division multiple access (CDMA) base stations. Since 2004, he has been a Professor with the Department of Electronic Engineering, Myongji College, Seoul. His main research interests include radio resource management, interference mitigation techniques, multiple-input multiple-output (MIMO) systems, and spectrum engineering.



KYUNG-YUL CHEON received the B.S. degree in electrical engineering from Korea University and the M.S. degree in electrical engineering from Seoul National University, Seoul, South Korea. He is currently a Principal Member of the Engineering Staff with the Electronics and Telecommunications Research Institute, South Korea. His current research interest includes mobile data measurement and analysis for spectrum management.



YONG-HOON CHOI (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Yonsei University, Seoul, South Korea, in 1995, 1997, and 2001, respectively.



SEUNGKEUN PARK (Member, IEEE) received the B.S. and M.S. degrees in applied statistics from Korea University, Seoul, South Korea, in 1991 and 1993, respectively, and the Ph.D. degree in information communication engineering from the University of Chungbuk, Cheongju, South Korea, in 2004. He is currently an Assistant Vice President with the Electronics and Telecommunications Research Institute, Daejeon. His current research interests include communication theory and spectrum management.

From 2001 to 2002, he was a Research Associate with the Institute for Systems Research (ISR), University of Maryland, College Park, MD, USA. From 2002 to 2005, he was a Chief Research Engineer with LG Electronics. Since 2005, he has been working with the Faculty of Kwangwoon University, Seoul, South Korea. His research interests include communication network software development, positioning technology, and speech recognition and synthesis. He received the 28th Choon-Gang Award in 2013 from the Choon-Gang Memorial Association.

...